SYSTEM AND METHOD FOR DETERMINING UNMATCHED DESIGN ELEMENTS IN A COMPUTER-AUTOMATED DESIGN

RELATED APPLICATIONS

[0001] The present document contains material related to the material of copending, cofiled, U.S. patent applications Attorney Docket Number 100111221-1, entitled System And Method For Determining Wire Capacitance For A VLSI Circuit; Attorney Docket Number 100111227-1, entitled System And Method For Determining Applicable Configuration Information For Use In Analysis Of A Computer Aided Design; Attorney Docket Number 100111228-1, entitled Systems And Methods Utilizing Fast Analysis Information During Detailed Analysis Of A Circuit Design; Attorney Docket Number 100111230-1, entitled Systems And Methods For Determining Activity Factors Of A Circuit Design; Attorney Docket Number 100111232-1, entitled System And Method For Determining A Highest Level Signal Name In A Hierarchical VLSI Design; Attorney Docket Number 100111233-1, entitled System And Method For Determining Connectivity Of Nets In A Hierarchical Circuit Design; Attorney Docket Number 100111234-1, entitled System And Method Analyzing Design Elements In Computer Aided Design Tools; Attorney Docket Number 100111236-1, entitled Computer Aided Design Systems And Methods With Reduced Memory Utilization; Attorney Docket Number 100111238-1, entitled System And Method For Iteratively Traversing A Hierarchical Circuit Design; Attorney Docket Number 100111257-1, entitled Systems And Methods For Establishing Data Model Consistency Of Computer Aided Design Tools; Attorney Docket Number 100111259-1, entitled Systems And Methods For Identifying Data Sources Associated With A Circuit Design; and Attorney Docket Number 100111260-1, entitled Systems And Methods For Performing Circuit Analysis On A Circuit Design, the disclosures of which are hereby incorporated herein by reference.

BACKGROUND

[0002] In a circuit consisting of P– and N-type FETs (Field Effect Transistors), a condition called a 'drive fight' can exist when both a P-FET and an N-FET that share a common net connection are turned on, thus creating a path from a voltage rail (e.g., VDD) to ground. This drive fight is undesirable from a power consumption standpoint, since it means that current essentially flows directly from the positive supply rail to the negative one through the FETs, and is therefore a waste of current. Both the P-FET and N-FET may be a stack of channel connected FETs, as long as the stacks share a common node. One way to avoid drive fights is to use complementary FET logic, which means that for every P-FET in the logic gate, there is an N-FET that shares the same input signal, such that when the P-FET is on, the N-FET is off, and vice versa. This complementary FET logic prevents drive fight conditions. However, complementary (e.g., CMOS) logic can be slow, so 'dynamic logic' may be employed to enhance circuit performance. Dynamic logic basically consists of a P-FET that pulls the output of the logic gate high, and some sort of tree of N-FETs connected to the P-FET that can pull the node low if necessary. A problem with dynamic logic is that if there is no corresponding N-FET in each branch of the tree for each P-FET, then a drive fight condition results. Therefore, what is needed is a method for identifying situations where a drive fight can occur, and also for determining the amount of current that the drive fight sinks.

SUMMARY

[0003] The present system attempts to find situations where a drive fight can occur in a circuit, and determines the relative area of FETs that are drive fighting, to determine the amount of current that the drive fight sinks. The system determines instances of a first type and a second type of the design elements that are connected to a specific node in the circuit, and stores the gate signal name for each determined occurrence of the first type of design element in a first list. The gate signal name for each determined occurrence of the second type of design element is than stored in a second list. A value of a design element characteristic and indicia thereof for each determined occurrence of the first and the second types of the design elements is than stored. A set difference operation is preformed on the first list and the second list to

determine orphan gate signal names that appear in only one said list; and a cumulative value is determined for each said design element characteristic, by adding the design element characteristic value corresponding to each stored said orphan gate signal name, to produce a total design element characteristic value.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] Figure 1 shows an exemplary CAD system configured for determining unmatched design elements in accordance with one embodiment of the present method; and

[0005] Figure 2 is a flowchart illustrating an exemplary set of steps performed in the embodiment of the system of Figure 1.

DETAILED DESCRIPTION

[0006] Figure 1 shows a CAD system 100 configured for analyzing a plurality of design elements 111 of a design 109. As shown in Figure 1, CAD system 100 includes computer system 101 and CAD tool 107. Computer system 101 controls CAD tool 107 to analyze design 109. Computer system 101 includes processor 102, computer memory 104, and storage unit 106. In computer system 101, processor 102 is coupled to storage unit 106 for loading CAD tool 107 into computer memory 104. Design 109, or a part thereof, is loaded in computer memory 104 upon initialization of CAD tool 107. Computer system 101 detects occurrences of first and second types of a design elements 111 connected to a particular node, and determines the net difference of a desired electrical characteristic with respect to the two design element types connected to the node.

[0007] A design element is a component of design 109. The first and the second types of design elements may be from any family of design elements used in the design of interest, with both types being from the same family. For example, an electronic design may include design elements such as transistors, wires, resistors, capacitors, power sources, and logic gates. The first and the second types of design elements may respectively be, for example, P-type and N-type MOSFETs from a design element family of transistors. A design element family is a set of two or more design elements, each of which has at least one common electrical or physical characteristic. For example, a P-type MOSFET and an N-type MOSFET may be

considered as two members of a design element family. Both types of MOSFETS have common characteristics, such as MOSFET widths (explained below). In an exemplary embodiment, for a given analysis, a design element family is predetermined to consist of two design elements, each of which corresponds to the other with respect to some electrical characteristic. Therefore, an unmatched design element is a design element that does not have a corresponding design element within the design, such as a P-type MOSFET with no corresponding N-type MOSFET.

[0008] In the present system, processor 102 is configured for determining the difference between a cumulative value of a predetermined characteristic of unmatched first type of design elements connected to a given node.

[0009] In operation, processor 102 determines an applicable value of the desired characteristic of the design elements 111 and cumulatively stores the value for later calculations. The characteristic may represent, for example, a dimension of a particular design element. In an exemplary embodiment, the width of each unmatched particular type of MOSFET (e.g., a P-type MOSFET that has no corresponding connection to an N-type MOSFET) is determined for the unmatched instances of that type of MOSFET. The width of a MOSFET is a factor that determines the overall area of the MOSFET, and thus may be used, for example, to calculate a total drive fight current for a particular circuit design.

[0010] Figure 2 is a flowchart illustrating an exemplary set of steps performed in one embodiment of the present system. As shown in Figure 2, processor 102 initially selects a 'PN' node in the design 109 to be analyzed, at step 205. A PN node is a node with a connection between a P-FET and an N-FET. In steps 210 – 230, each instance of a P-FET connected to this PN node is found and processed by iterating over all of the design elements 111 connected to the selected node.

[0011] More specifically, for each P-FET found at step 210, the gate signal name of the P-FET is added to a first list 110 in computer memory 104, at step 215, to create a set of names of gate signals connected to P-FETs. At step 220, processor 102 determines a FET source current (the amount of current that the FET could source), by multiplying the width of the FET by a current density figure. The FET width data is part of the circuit information that exists in the data model that represents the circuit, and current density data is provided by a technology file for the

process in which the chip is to be constructed. At step 225, this source current value is stored cumulatively (i.e., as a total current value) in a source current hash table 105 that associates a total current value with each P-FET gate signal name. In the present embodiment, entries are stored in hash table 105 with a key of the gate signal name and a value of the corresponding FET source current. A hash table 105 is a 'lookup' table, known in the art, that is used to facilitate the storage and retrieval of data. It should be noted that other data storage/retrieval methods and structures known in the art may, alternatively, be used to perform the function of hash table 105, by storing a characteristic value (e.g., a FET source current value) and corresponding indicia, for each signal gate name. At step 230, if all P-FETs connected to this node have not been processed, then processing continues at step 210.

[0012] Once all P-FETs connected to this node have been processed, then in steps 235 – 245, each instance of a N-FET connected to this PN node is found and processed by iterating over all of the design elements 111 connected to the node. More specifically, for each N-FET found at step 235, the gate signal name of the N-FET is added to a second list 112 in computer memory 104, at step 240, to create a set of names of gate signals connected to N-FETs. At step 245, if all N-FETs connected to the selected node have not been processed, then processing continues at step 235.

[0013] Once all N-FETs connected to the present node have been processed, then at step 250, a set-difference operation is performed on the two sets of gate signal names stored in lists 110 and 112, to find the names of the signals that appear in one set or the other, but not in both. These signal names denote 'orphan' signals.

[0014] At steps 255 – 265, a P-FET gate signal name is found for every orphan signal in source current hash table 105, and, for each of the orphan gate signal names, the source current value corresponding to the gate signal name in the hash table is added to a cumulative current value, to produce a total source current value. This total source current value represents the total drive fight current that may be sourced in this logic gate. A scaled current value may then be calculated by multiplying the total drive fight current by a scale factor indicating the length of time (as a percentage of the applicable clock cycle) that this PN node is likely to drive fight. This scaled current value represents the drive fight current on this PN node.

[0015] While the process shown in Figure 2 illustrates an algorithm for determining unmatched P-FET current using single element N-FET stacks, the present method and system can be extended to multi-element N-FET stacks. In a multi-element N-FET stack embodiment, after steps 210 –230 of Figure 2 are performed, an iteration is performed over all the N-FETs connected to the PN node (as shown in steps 235 – 245 of Figure 2), and then an iteration is continued over N-FETs following 'channel-connected' FETs, until a GND (ground) connection is encountered. Channel-connected FETs are FETs that are connected at the source or drain to another FET's source or drain terminal. The current path from source to drain of a FET is referred to as a 'channel', and thus two FETs that are connected at source and drain may be considered to be 'channel-connected.' The process is fundamentally similar to that of Figure 2, in that the N-FET gate signal is cumulatively added to the set of N-FET gate signal names, and then the set difference operation (step 250 in Figure 2) is performed on the two sets of gate signal names stored in lists 110 and 112. Orphan gate signals are then processed as shown in step 255 – 265 of Figure 2.

[0016] Instructions that perform the operations described with respect to Figure 2 may be stored in computer memory or other computer-readable media, and later retrieved therefrom and executed by processor 102 to operate in accordance with the present system. Examples of instructions include software, program code, and firmware. Examples of storage media include memory devices, tapes, disks, integrated circuits, and servers.

[0017] Changes may be made in the above method and system without departing from the scope hereof. It should thus be noted that that the matter contained in the above description or shown in the accompanying drawings should be interpreted as illustrative and not in a limiting sense. The following claims are intended to cover all generic and specific features described herein, as well as all statements of the scope of the present method and system, which, as a matter of language, might be said to fall there between.